

+

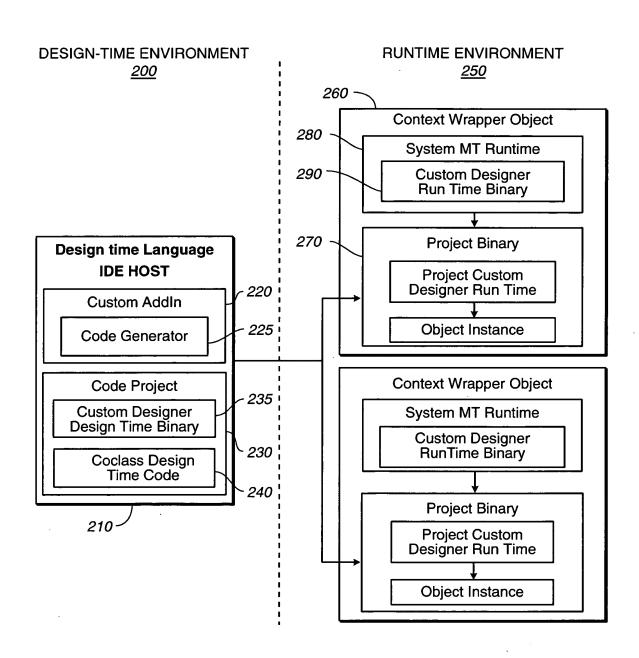
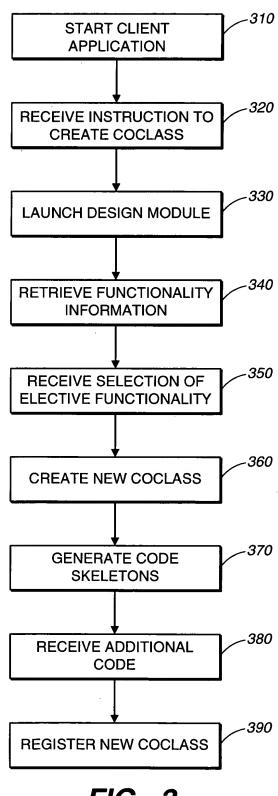


FIG._2

+



300

FIG._3

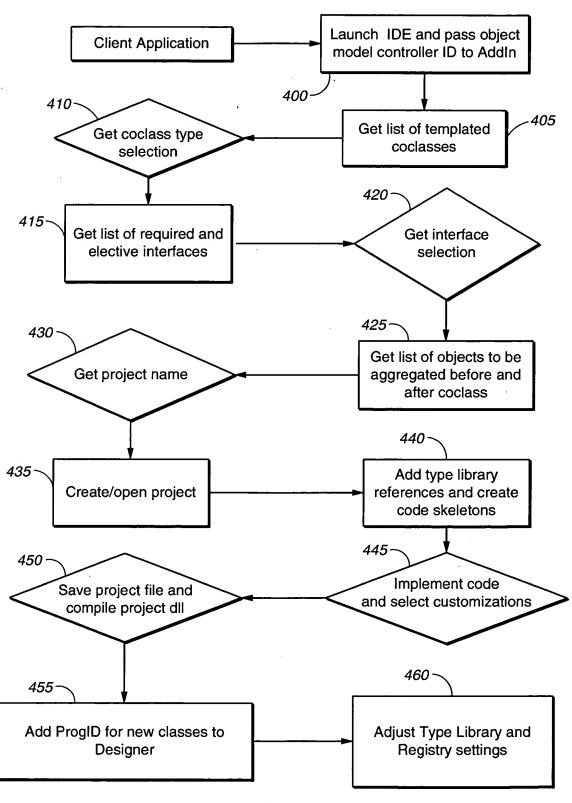
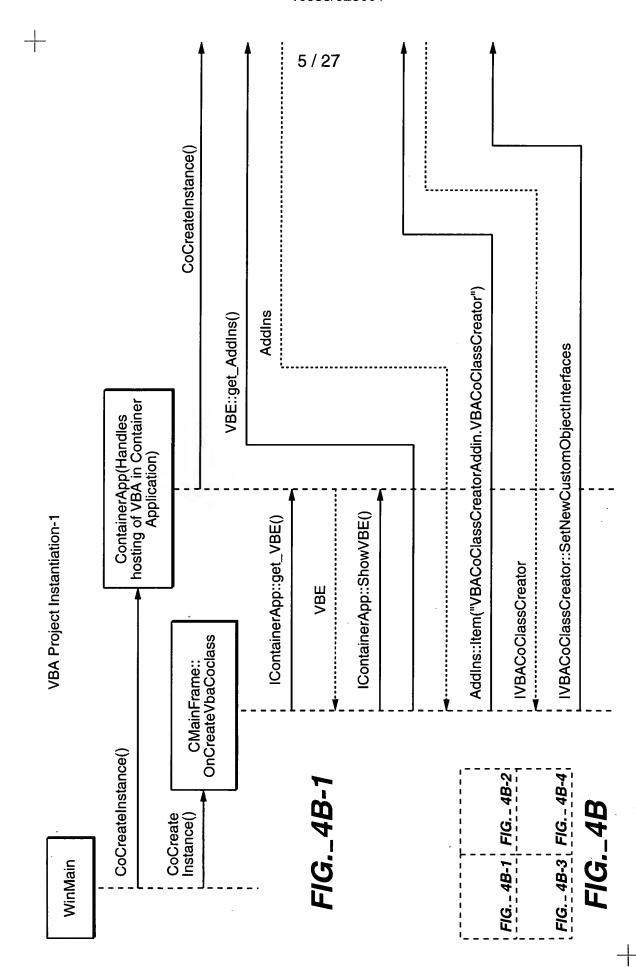
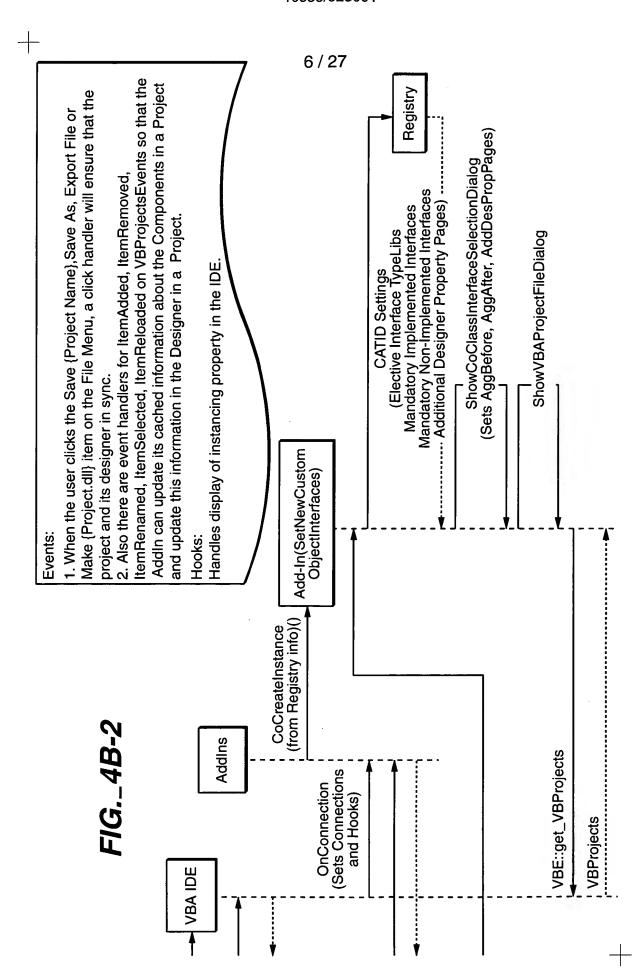


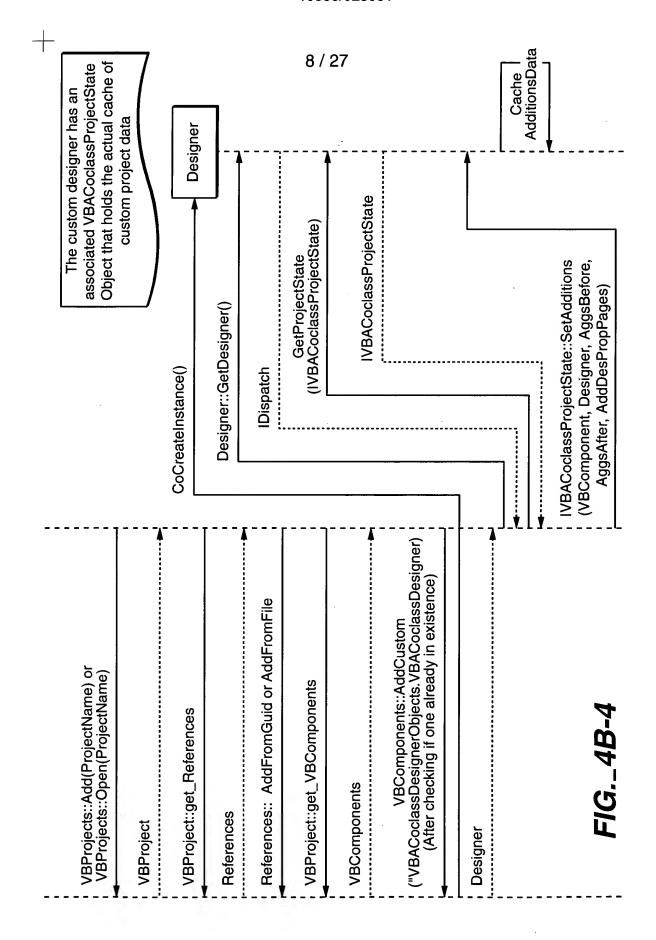
FIG._4A



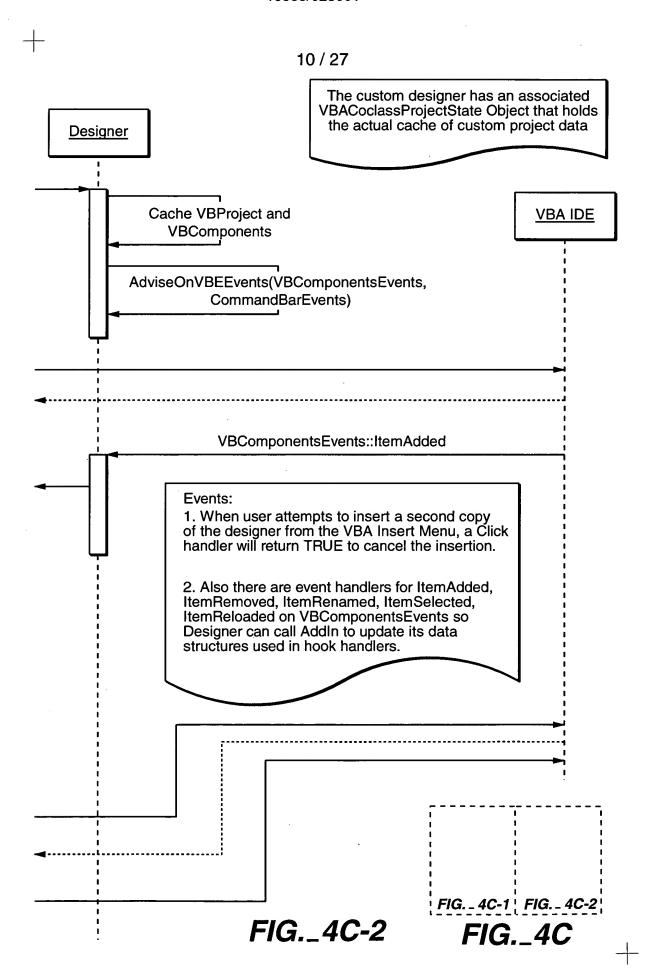


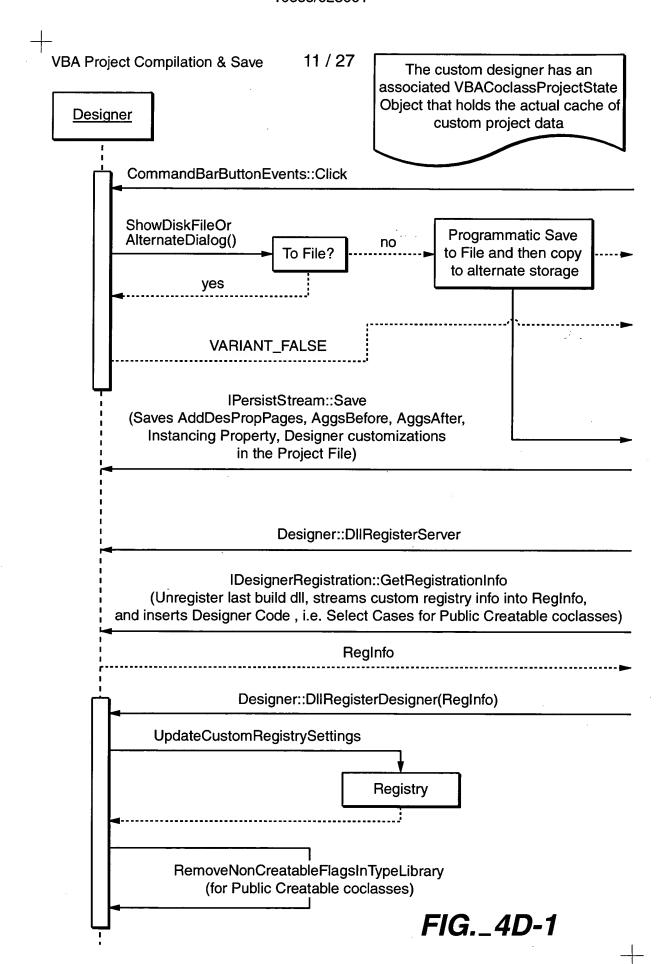
To allow alternate storage such as to a database for now it will be necessary to persist the project file to a disk file first because the MT implementation only allows persistence in a disk file.

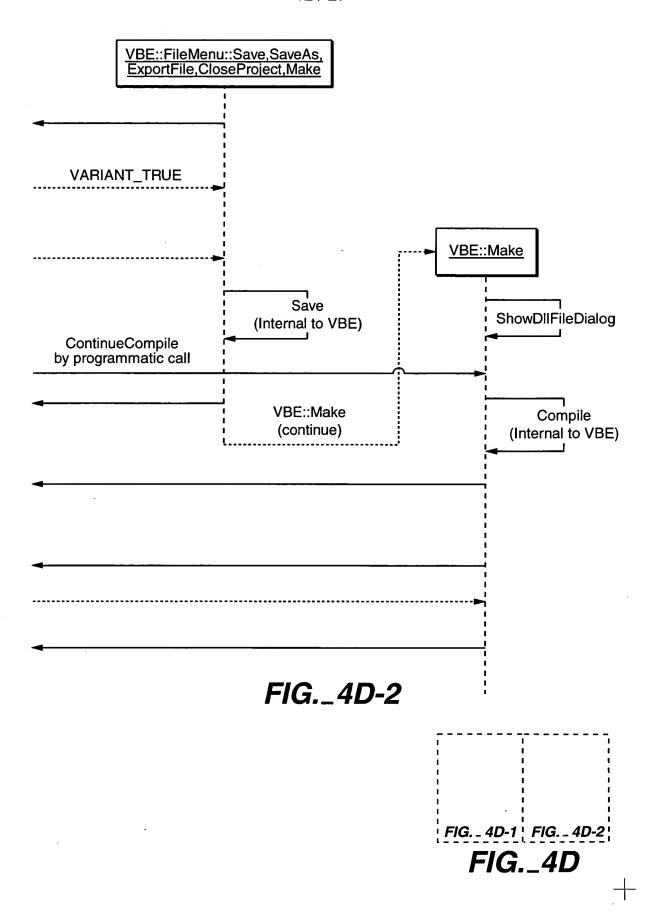
FIG._4B-3

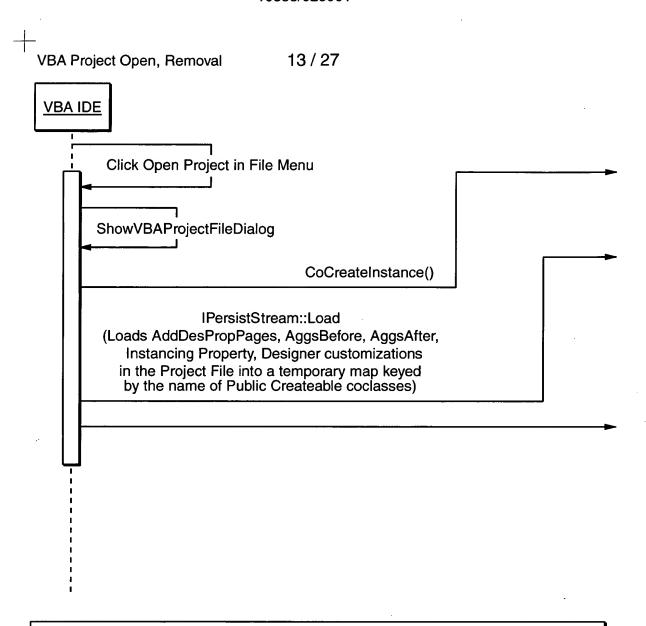


9/27 VBA Project Instantiation-2 Add-In(SetNewCustomObject IVBACoclassProjectState::AttachStateObject Interfaces continued) (VBComponent(VBA coclass), VBComponent (VBA designer), IManageVBACoclassDesigner, **VBProject**, VBComponents) VBComponents::Add (vbext_ct_ClassModule) **VBComponent** IIVBACoclassCreator::ComponentAdded (IManageVBACoclassDesigner, VBProject, VBComponent(VBA coclass), InitialInstancing PropertyValue) (Info used in hook handlers) GenerateInterfaceCode(IIDs) IVBACodeGenerator:: GenerateVBACode(IID) for each IID **VBACodeGenerator CATIDs** VBComponent::get_CodeModule CodeModule CodeModule::InsertLines(Code) FIG._4C-1





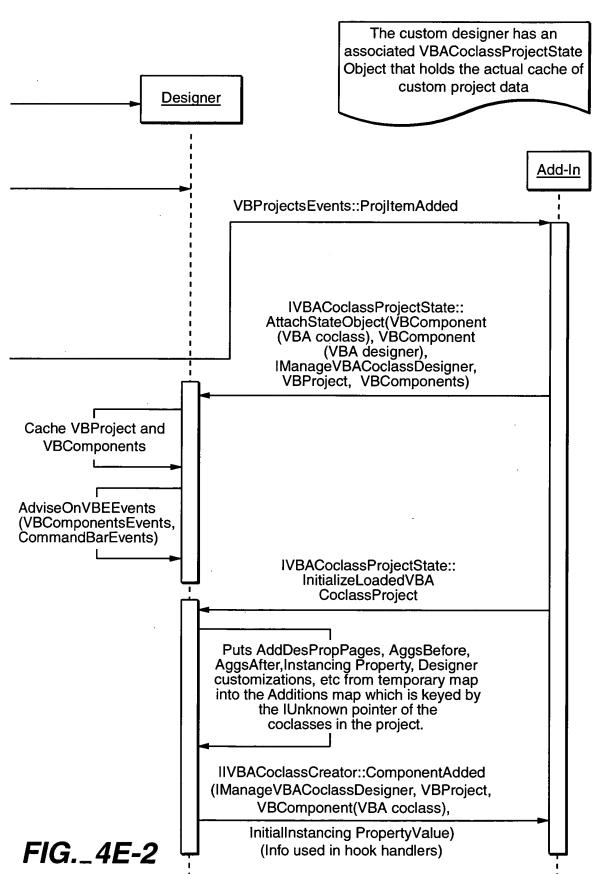




When the user selects deleting a project from the VBA IDE, the VBProjectsEvents:: ProjltemRemoved handler in the Addin removes the project's custom settings in the data structures in the Addin. Also when the user inserts/deletes a designer into/from a project, or inserts/deletes a class into/from a project, the VBCommandBarEvents:: Click event handler in the Addin and designer, respectively, add to or delete from the data caches in the Addin and designer.

FIG._4E-1

FIG._4E-1 | FIG._4E-2 | FIG._4E



 \perp

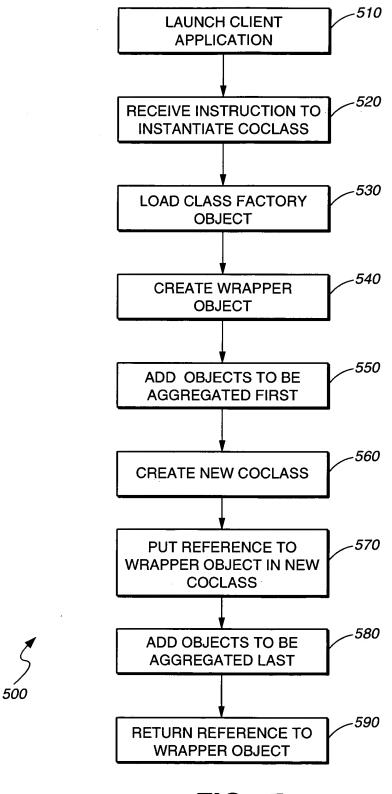


FIG._5

16/27 **VBA** Object Instantiation DilGetClassObject caches the Coclass VBACoclassWrapperObj::FinalConstruct CLSID and returns the class factory for the Wrapper object. Client VBAWrapperObj (FinalConstruct) CoCreateInstance (CoclassCLSID) CoclassCLSID Registry **DesClassFactoryProgID** CoclassCLSID Registry AggsBeforeCLSIDs, AggAfterCLSIDs CreateAggsBefore and AddToAggObjectVector CoCreateInstance() IDispatch::QueryInterface (ISymyxVbaClassFactory) ISymyxVbaClassFactory AddToAggObjectVector (VBACoclass) AddCoclassToAggObjectVector and CreateAggsAfter and AddToAggObjectVector FIG._6A | FIG._6B

FIG._6A

FIG._6

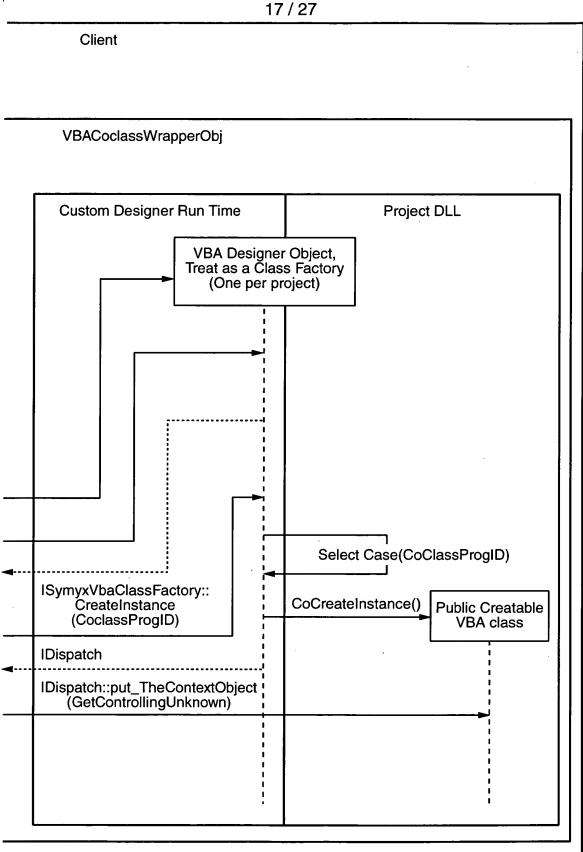


FIG._6B

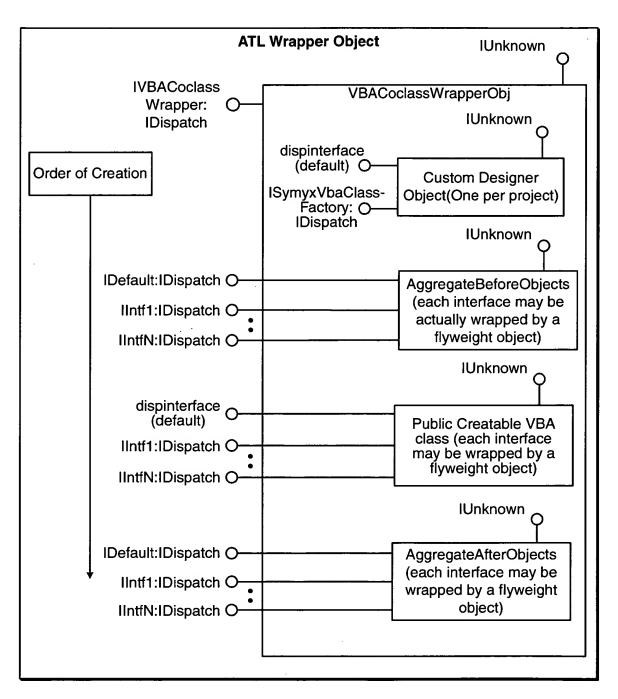
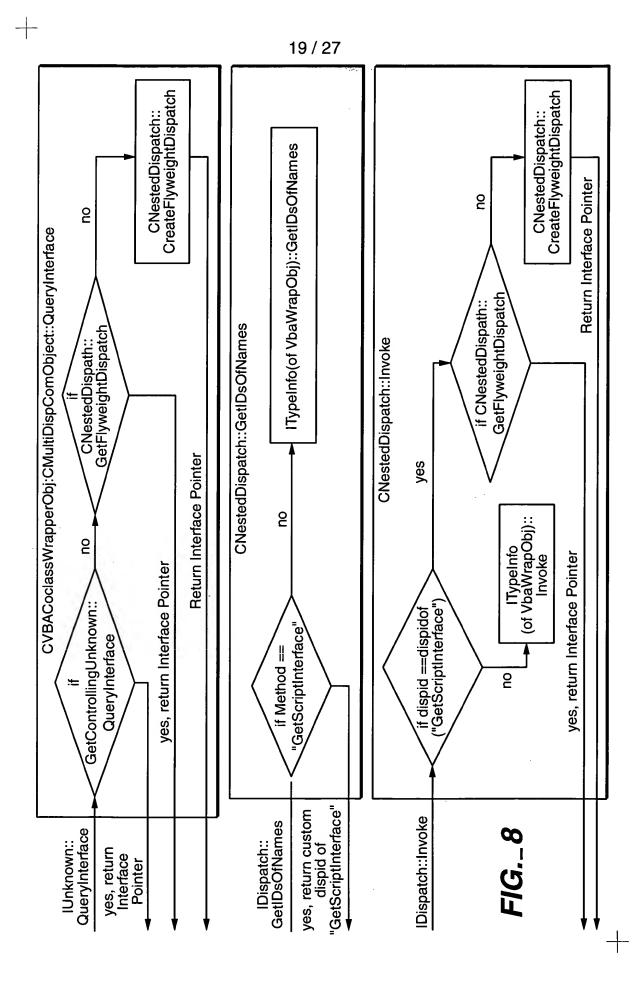


FIG._7



20 / 27

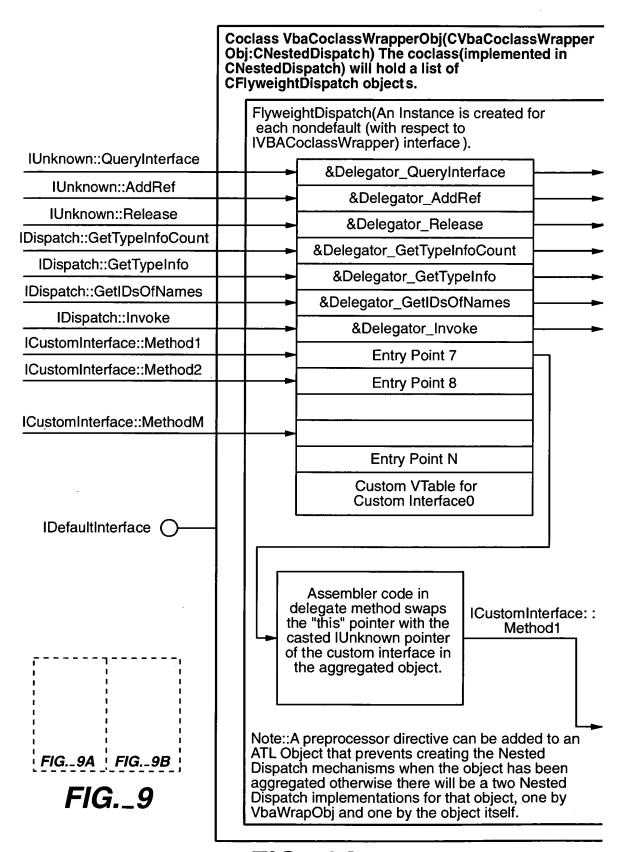


FIG._9A

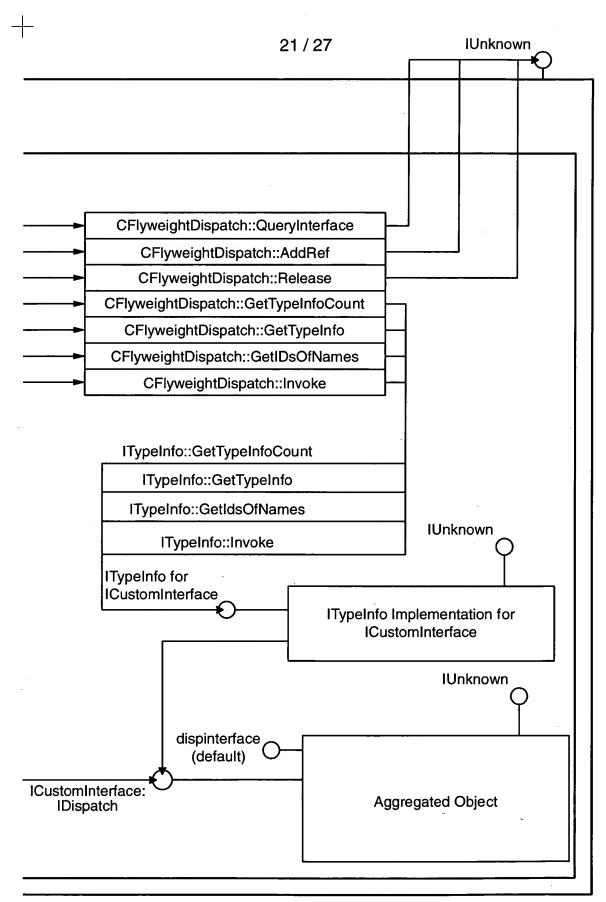
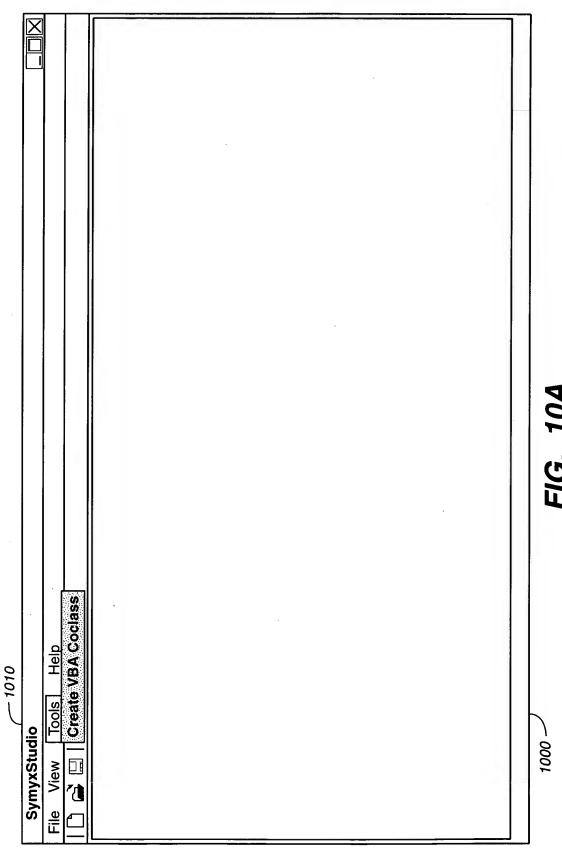
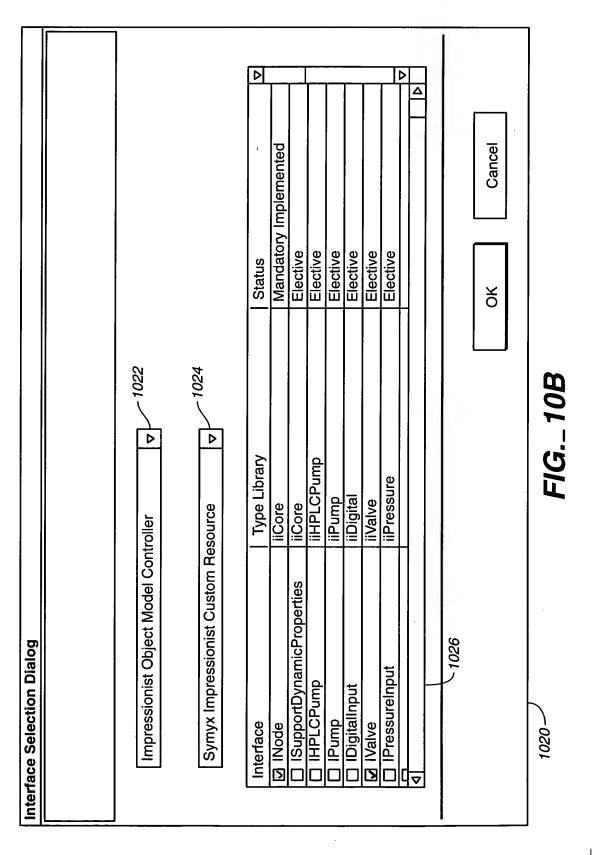


FIG._9B

22 / 27



23 / 27



+

Microsoft Visual Basic - Project1 Microsoft Visual Basic	
File Edit View Insert Format Debug	Run Tools Add-Ins Window Help
4 > ♂ 日 📵 目 ※ 🖂 ^ 🖺 🗏	
Project1 - CustomValveProject x	
	Project1 - VBACoclassDesigner1 (VBACoclassDesinger)
日本CustomValveProject (Project1)	৺Project1 - Class1 (Code)
Glass Modules	IValve ∇ Wait ∇
Class1	Option Explicit
⊕⊟ Designers	Implements IValve
	Private Sub Class_Initialize()
Properties - Class1 x	
Class1 ClassModule	Frivate Sub Class_Terminate() End Sub
Alphabetic Categorized	
(Name) Class1	Private Property Get IValve_Positions() As Long End Property
Instancing 3 - PublicCreatable	Driveta Dronarty Lat TVF Docition (BAM)
	End Property
	Private Property Get IValve_Position() As Long1032
	Private Sub IValve_Iniialize() \to 1032 End Sub
	Private Sub IValve_Wait()

FIG._ 10C

1030

			[F	<u></u>			마	Þ									D				
				訓	∦ 1⊅			Γ									<u>_</u> [D	
/── Microsoft Visual Basic - F:\VBAIntegrationDemo\Custom\ValveProject.vba	Eile Edit View Insert Format Debug Run Tools Add-Ins Window Help	数 	ıstomValveProject.vba - Class1 (Code) □□ ×	∇ Initialize ∇		As Long	m_Address As IPropertyInteger m_NumPositions As IPropertySinglecChoice	alize()	As INode = TheContextObject ion = 1 ResID = 0 = "Dave's Custom Valve"		Set m_Address = New PropertyInteger m_Address.Name = "Address" m_Address.XMLTag = "Address" m_Address.Valve = 1 AggNode.Properties.Add m_Address		ns = New PropertyEnum	Set m_NumPositions = New PropertyEnum m_NumPositions.Name = "Number of Positions" m_NumPositions.XMLTag = "NumPositions" m_NumPositions.Choices = "2 4 8 16" m_NumPositions.Text = "2" 4 8 16" AggNode.Properties.Add m_NumPositions				IValve_Positions() As Long ons = Cing(m_NumPositions.Text)	Property let IValve Position(ByVal newValue As Long) m_ValvePosition = newValue	IValve Position() As Long n = m_ValvePosition	Δ
			►:\VBAIntegrationDemo\CustomValveProject.vba - Class1	Class	Implements IValve	Private m_ValvePosition Private m_Address As IP Private m_NumPositions	Private Sub Class_Initialize()	Dim AggNode As INode Set AggNode = TheCont AggNode.Version = 1 AggNode.IconResID = C AggNode.Name = "Dave"	Aggnode. LconRes Aggnode. Name = '	Set m_Address = m_Address.Name =	M_Address.Valve AddNode.Propert	Set m NumPositic	Set m. NumPositic m. NumPositions. N m. NumPositions. C m. NumPositions. C m. NumPositions. C AggNode. Properti		m_ValuePosition	m_ValuePosition End Sub	Private Property Get IV IValve Positions End Property	Private Property let IV m_ValvePosition End Property	ate Property Get IValve_Positio	□	
			Project1 - CustomValveProject x		☐ ♦ CustomValveProject (F:\VBAIntegration Demo\Cust	⊕ ⊕ Modules	TheContextObjectModule	Class Modules	⊕ Designers		Properties - Class1	lle lle	Alphabetic Categorized	(Name) Class1	Instancing 3 - PublicCreatable						

FIG._ 10D

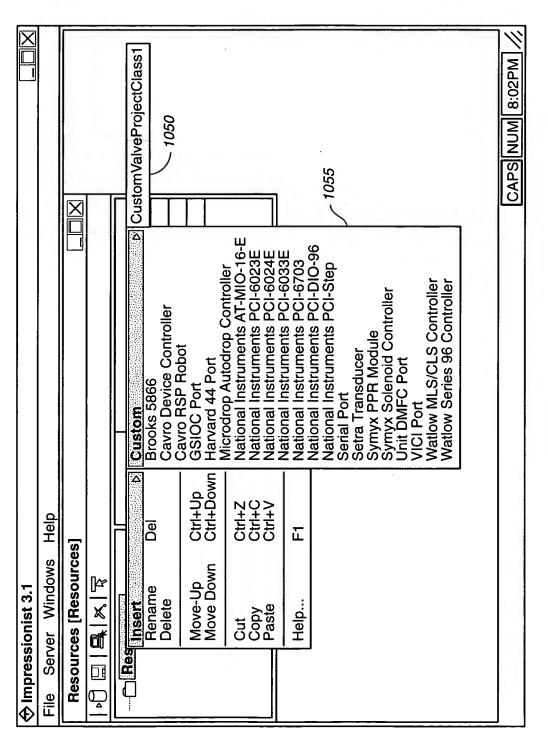


FIG._ 10E